

DAM
Desarrollo de Aplicaciones Multiplataforma
2º Curso

AD
Acceso a Datos

UD 0
Iniciación a Java

IES BALMIS
Dpto Informática
Curso 2019-2020
Versión 1 (09/2019)

UD0 – Iniciación a Java

ÍNDICE

13. Java en Linux

14. Geany

15. NetBeans

13. Java en Linux

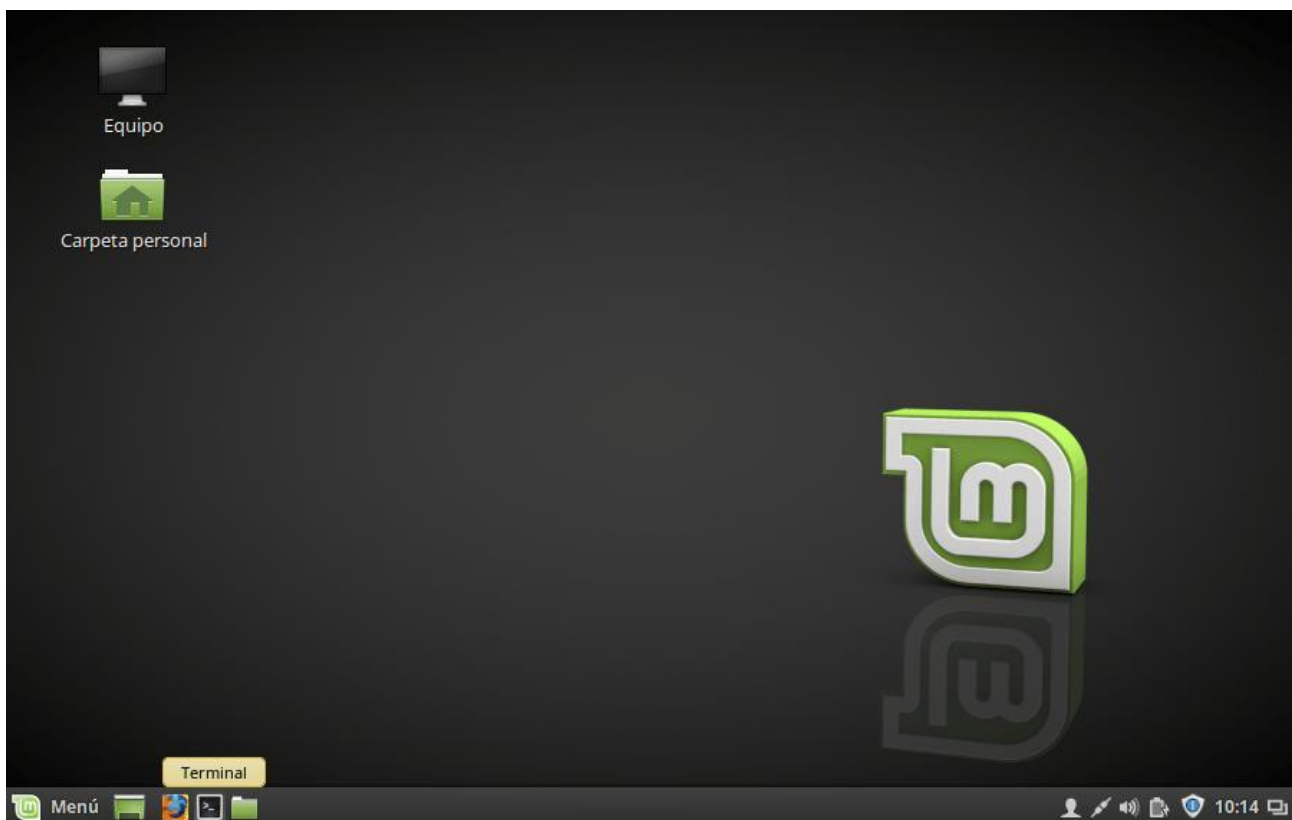
Como ya hemos comentado, para compilar los pasos son sencillos, aunque en la práctica se pueden complicar un poco:

1. Teclear el fuente usando cualquier editor de texto.
2. Compilar con “javac”, por ejemplo:
`javac HolaMundo.java`
3. Lanzar con “java”, por ejemplo:
`java HolaMundo`

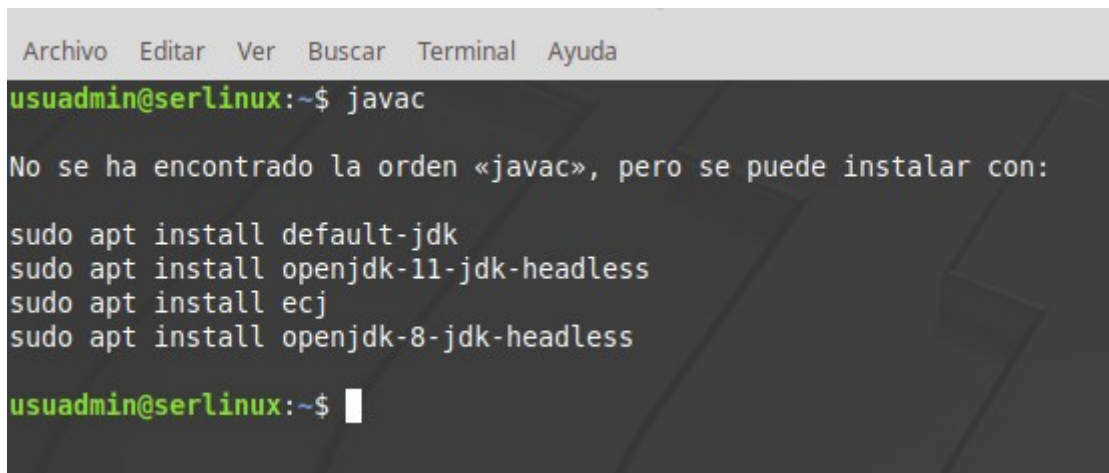
En la práctica pueden aparecer distintos problemas, como no tener Java instalado, que el nombre del fichero sea incorrecto o (en programas más complejos) no encontrar alguna de las bibliotecas que el programa necesite.

El problema más habitual, y el único que nos interesa por ahora, es que no esté instalado Java. Nos centraremos en una solución para el caso de Linux, en concreto con un Linux Mint 18.2 Cinnamon, que se basa en Ubuntu pero tiene un entorno de escritorio más “normal”.

En entornos un poco más convencionales, como Linux Mint con el escritorio Cinnamon, es habitual que haya un acceso directo al terminal directamente en la barra inferior:



El siguiente paso sería probar a lanzar el compilador de Java, tecleando “javac”. Es habitual que, en caso de que éste no se encuentre instalado, se nos muestren pistas de cómo instalarlo:



```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
usuadmin@serlinux:~$ javac

No se ha encontrado la orden «javac», pero se puede instalar con:

sudo apt install default-jdk
sudo apt install openjdk-11-jdk-headless
sudo apt install ecj
sudo apt install openjdk-8-jdk-headless

usuadmin@serlinux:~$
```

Como se nos indica, lo podemos instalar con “**sudo apt-get install default-jdk**”. Al final de la instalación podremos lanzar el editor “gedit” seguido del nombre del programa que deseamos crear o editar, como “**gedit HolaMundo.java**”. Si ese editor no está instalado, podemos optar por usar el del sistema (que quizá no realice la sintaxis en colores) o por instalarlo del mismo modo.

En la ventana de “gedit” podremos escribir el programa. Como ya le hemos indicado que se trata de un programa en Java, se mostrará la sintaxis realzada en colores, para ayudarnos a detectar fallos:

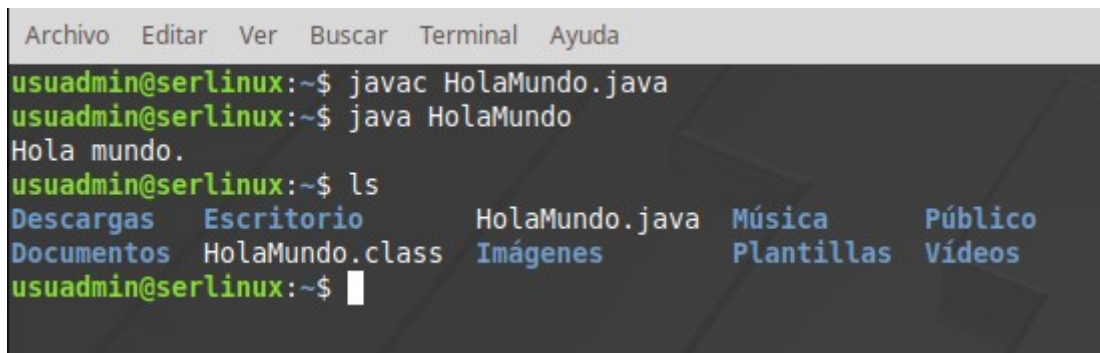


```
Abrir  [Icono]  *HolaMundo.java  Guardar  -  +  x
Archivo  Editar  Ver  Buscar  Herramientas  Documentos  Ayuda
// Fichero HolaMundo.java
// Compilar con: javac HolaMundo.java
// Ejecutar con: java HolaMundo

public class HolaMundo {
    public static void main(String [] args) {
        System.out.println("Hola mundo.");
    }
}
```

Si, por el contrario, hubiéramos abierto directamente "**gedit**" (o cualquier otro editor de textos) sin indicarle nombre de ficheros, es probable que la sintaxis no se muestre en colores, como mínimo hasta que guardemos el fichero con un nombre terminado en **".java"**.

Cuando salgamos del editor ya podremos usar la orden **"javac"** para compilar el programa y **"java"** para lanzarlo. La ejecución de "javac" nos creará el archivo **HolaMundo.class** que es que ejecutaremos con "java".



```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
usuadmin@serlinux:~$ javac HolaMundo.java
usuadmin@serlinux:~$ java HolaMundo
Hola mundo.
usuadmin@serlinux:~$ ls
Descargas  Escritorio  HolaMundo.java  Música  Público
Documentos  HolaMundo.class  Imágenes  Plantillas  Vídeos
usuadmin@serlinux:~$
```

14. Geany

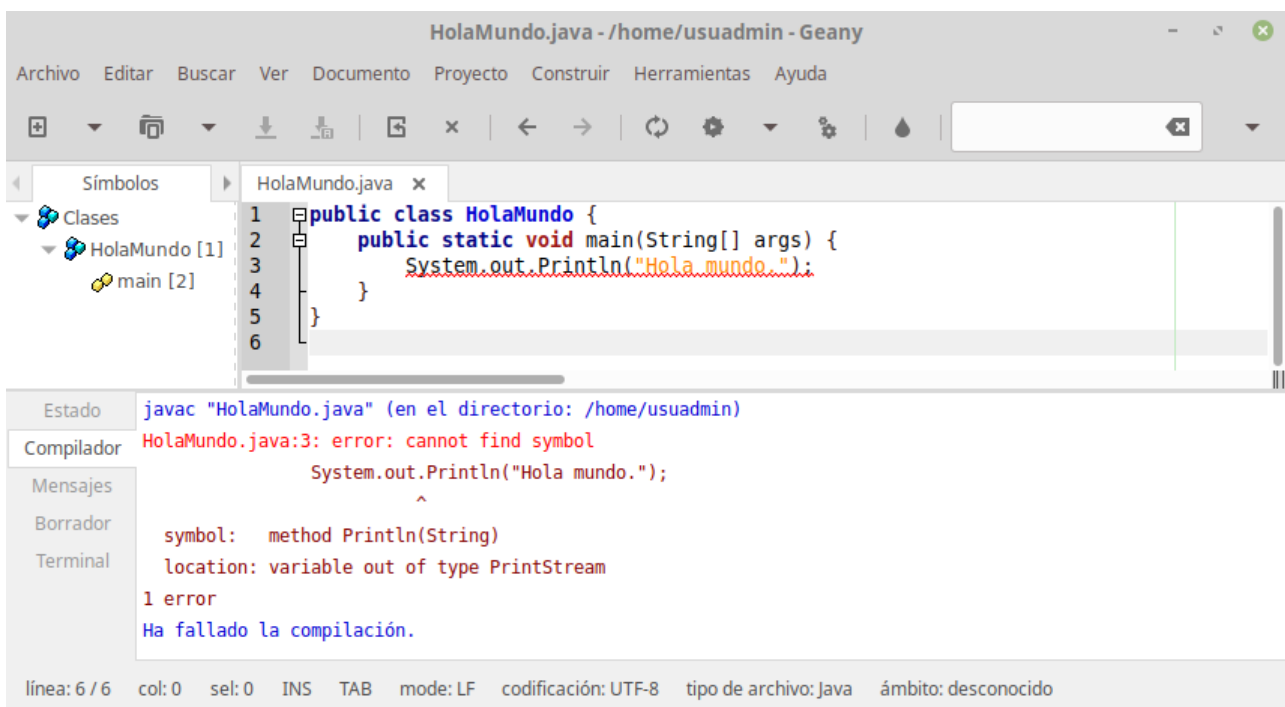
Hay alternativas más avanzadas, que permiten ahorrar algo de trabajo al no tener que teclear siempre "**javac**" y "**java**", pero que no suelen venir preinstaladas en el sistema.

Es el caso de **Geany**, un editor más potente que "**gedit**" con licencia GNU GPL (software libre y gratuito), y que permite compilar y lanzar el ejecutable simplemente con un clic. También permite saltar a las líneas que tienen un cierto error, en caso de que el programa no haya compilado correctamente. Es un entorno muy adecuado para programas sencillos, formados por un único fuente.

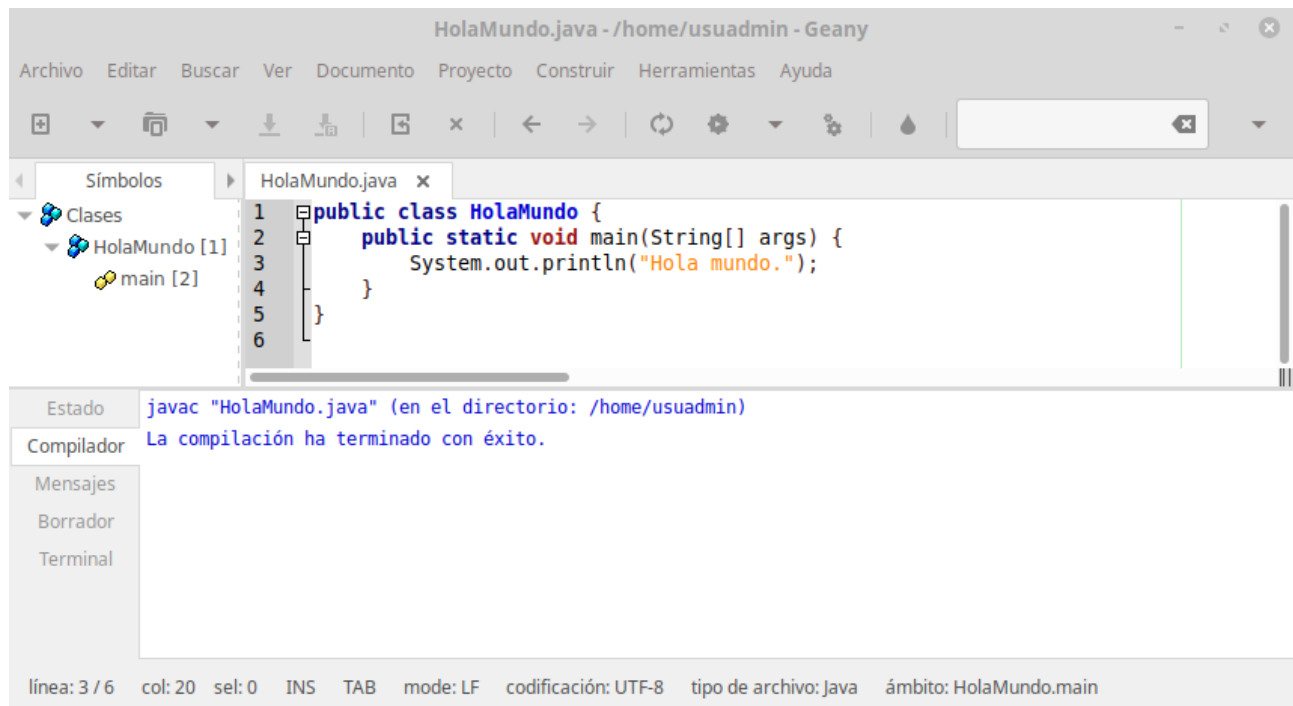
Para instalarlo en Windows bastaría con descargar el programa de su web oficial <https://www.geany.org/> y ejecutarlo.

Para instalarlo en Linux se puede ejecutar "**sudo apt-get install geany**".

Por ejemplo, ésta sería la apariencia de Geany tras intentar compilar un programa con errores:



Y ésta tras compilar un fuente sin errores:



15. Netbeans

NetBeans, al igual que otros IDE como **Eclipse**, es de mucho mayor tamaño, pero mucho más adecuado para programas formados por varios fuentes, ya que permite crear y mantener proyectos con facilidad.

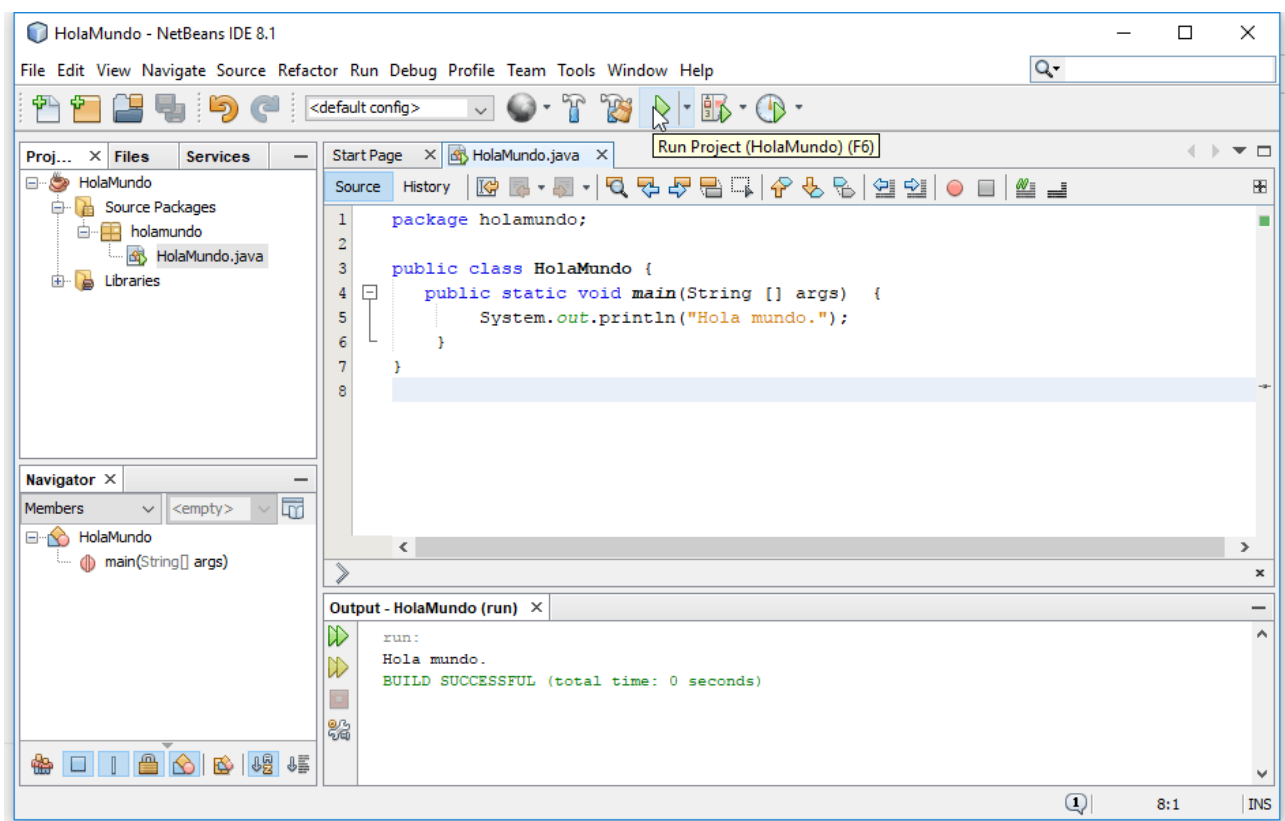
Para instalarlo en Windows accederemos al portal oficial para descargar la última versión 8.2 pre-apache:

<https://netbeans.org/downloads/8.2/>

Tiene varias ventajas frente a la versión 11 de Apache Netbeans:

- Está en **español**
- Si descargamos la **completa (ALL)**, tendremos todas las librerías instaladas

En el caso de usar (por ejemplo) NetBeans, bastaría con instalarlo, crear un nuevo proyecto (de tipo “Java Application”) y lanzarlo con el botón “Run Project”:



El programa creado desde NetBeans deberá formar parte de un "**package**", o no se podrá compilar correctamente:

```
package holamundo;

public class HolaMundo {
    public static void main(String [] args) {
        System.out.println("Hola mundo.");
    }
}
```

Un **package** (en español: “paquete”) es un contenedor de un grupo de clases, interfaces, etc. relacionadas entre si. Un paquete es a Java lo que una librería es a otro lenguaje (por ejemplo: C). Estos pueden contener clases, interfaces, enumerados, anotaciones e incluso otro tipo de archivos.

Las ventajas de organizar nuestros archivos en paquetes:

- Agrupar clases que tienen algo en común.
- Reutilización del código.
- Seguridad gracias a los modificadores de acceso.

El package se mostrará como una carpeta en el explorador del proyecto. Además